

# Computing segmentations directly from x-ray projection data via parametric deformable curves

Paper

**Dahl, Vedrana Andersen; Dahl, Anders Bjorholm; Hansen, Per Christian**

*Published in:*
Measurement Science and Technology

*Link to article, DOI:*
10.1088/1361-6501/aa950e

*Publication date:*
2018

*Document Version*
Peer reviewed version

# Computing Segmentations Directly From X-Ray Projection Data via Parametric Deformable Curves

**Vedrana Andersen Dahl, Anders Bjorholm Dahl, Per Christian Hansen**

Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark

E-mail: {vand,abda,pcha}@dtu.dk

June 2017

**Abstract.** We describe an efficient algorithm that computes a segmented reconstruction directly from X-ray projection data. Our algorithm uses a parametric curve to define the segmentation. Unlike similar approaches which are based on level-sets, our method avoids a pixel or voxel grid; hence the number of unknowns is reduced to the set of points that define the curve, and attenuation coefficients of the segments. Our current implementation uses a simple closed curve and is capable of separating one object from the background. However, our basic algorithm can be applied to an arbitrary topology and multiple objects corresponding to different attenuation coefficients in the reconstruction. Trough systematic tests we demonstrate a high robustness to the noise, and an excellent performance under a small number of projections.

*Keywords*: X-ray tomography, segmentation, tomographic reconstruction, deformable models, parametric curve

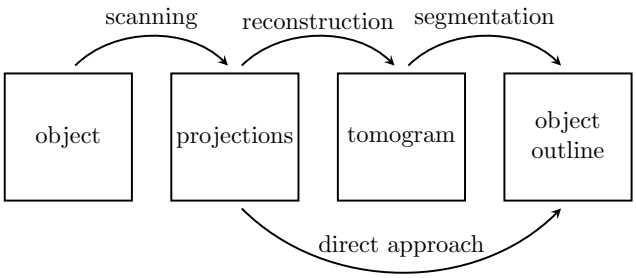


**Figure 1.** Two approaches to tomographic segmentation: the classical approach that involves an intermediate tomogram, and the direct approach which operates directly on the projection data.

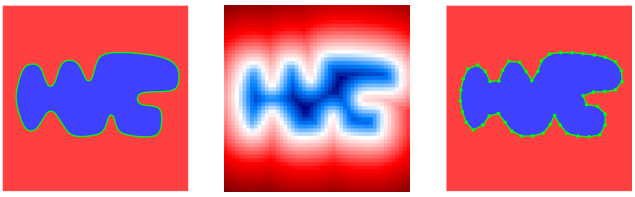

**Figure 2.** Implicit and parametric curve representation. *Left:* a simple curve divides the domain into inside and outside. *Middle:* the curve implicitly represented as a zero level-set of a signed distance field on a discrete $50 \times 50$ pixel grid. *Right:* the parametric curve represented using 50 points.

## 1. Introduction

Processing of X-ray tomographic projection data usually starts by computing a reconstructed 2D or 3D image – sometimes called the tomogram. Reconstruction is often followed by an image segmentation step, aiming at partitioning the tomogram into regions of approximately constant gray level.

Each of these processing steps introduces errors and artefacts, especially pronounced in case of noisy or incomplete data. This has motivated the development of various methods which combine the reconstruction and the segmentation step, see Figure 1. Examples of approaches relevant in our context include the DART algorithm [1] and fitting deformable curves to projection data [2]. Such methods are useful when the object under study consists of a number of domains with approximately homogeneous absorption coefficients.

### *1.1. Deformable Curves*

In this work we represent the segmentation of the reconstruction domain by means of deformable curves, which are already extensively used for segmentation of noisy images. Deformable curves come in two variants: *parametric* curves and *implicit* (also known as geometric) curves; see Figure 2 for illustrations. A parametric curve, in a discrete setting, is fully defined by a (small) sequence of points connected by line segments. A well known parametric curve is the snakes method introduced by Kass et al. [3]. Implicit curves are defined on a regular grid and use the level-set methodology by Osher and Sethian [4] for the deformation.

Implicit curves have gained popularity in tomography, possibly because of their ability to handle topology changes during the reconstruction, when the curve is fitted to the data. Parametric curves, on the other hand, also have their advantages – primarily that they do not rely on discretizing the reconstruction domain into a large number of pixels/voxels.

In this work, we use parametric deformable curves for tomographic segmentation, i.e., segmentation of the reconstruction domain, directly from the projection data. As such, we never consider a pixel or voxel representation of the scanned object, and hence we avoid the large number of unknowns in the traditional approach.

A parametric curve representation will be advantageous in applications where geometric representation of a scanned object is required. With a classical grid-based approach this involves reconstructing a tomogram, segmenting it into a binary image, and then meshing the result to obtain geometry – these steps are carried out independently, so errors can propagated to the later stages. When the geometric representation is used for e.g. appearance rendering, the precise geometry might be less important. But in an application like metrology, where an exact geometric representation is essential, there is a clear advantage of a precise geometry. Furthermore, the use of an explicit parametric curve will make the comparison to a reference model much easier.

Algorithms that involve deformable curves rely on a methodology for changing an initial curve, such that it eventually provides the desired result – here a segmented reconstruction whose X-ray projection fits the measured projection data.

A crucial component of a parametric deformable model is a force that evolves a curve towards the desired result. The definition and efficient computation of this force is the focus of this paper, and our contributions are twofold. In Section 2 we derive the deformation force for evolving a parametric curve using the projection data, such that we obtain the desired segmentation of the reconstruction domain. In Section 3 we systematically test the performance and robustness of our method in the presence of noise and a limited number of projections.

In this work we consider only a *single closed* curve, which suffices to illustrate our main points. However,

the forces we derive may be used for segmenting multiple objects of different attenuation and arbitrary topology.

### *1.2. Related work*

One of the most popular methods for image segmentation via deformable curves is active contour without edges, developed by Chan and Vese [5]. This method minimizes a piecewise constant Mumford-Shah functional [6] and uses level-sets for the curve representation.

Algorithms for tomographic reconstructions utilizing a piecewise constant model rely on minimizing a global energy in the projections space. This leads to curve deformation algorithms similar to ours, cf. [2] and [7], but only in combination with the level-set curve representation. A related 3D algorithm also based on level-sets is presented in [8].

Methods which combine level-sets with deformation forces not closely related to ours are suggested in [9], [10] and [11]. Examples of piecewise smooth models using level-sets are [12], [13] and [14], while multiphase approaches include [15] and [16].

There are a few examples [17] and [18] of using a parametric deformable curve for tomographic reconstruction. These methods attempt to locate the boundaries of the object by exploiting the discontinuities in the sinogram, while [19] handles both 2D and 3D segmentation of a single closed object with a forward model based on ray tracing.

A related mesh-based approach, but without boundary deformation, involves triangulating the reconstruction domain. Such per-triangle reconstruction with a content adaptive mesh is suggested in [20]. Further processing is suggested in methods from [21] and [22], where segmentation edges between the triangles are detected and iteratively refined. A similar approach for 3D reconstruction is [23].

Also related to our approach is discrete tomography [1] and its variants [24] and [25], where the pixel/voxel values are restricted to a small number of discrete values, each corresponding to a different phase in the object.

### *1.3. Contributions and outlook*

In this work we deform a parametric curve in the reconstruction domain to minimize an energy in the projection domain. As such, our method combines the minimization principle of level-set methods with the parametric curve. The use of parametric curves in tomography has been limited, and the strong dominance of level-set methods can probably be explained by a critical weakness of parametric curves, namely, the lack of topological adaptivity. Furthermore, level-set curves are attractive due to the mathematical rigor in deriving the deformation.

Many attempts were made to equip the parametric curve with topological adaptivity [26], but the suggested solutions were cumbersome and computationally costly. However, recent progress in interface tracking has led to parametric curve representations with an efficient solution to topological adaptivity. In particular, a method by Misztal et al. [27] (originally developed for fluid simulation [28]) already has an application in image segmentation [29] and may readily be used for 2D and 3D tomographic reconstruction. This exciting advance eliminates the biggest obstacle in using parametric curves, and is a main motivating factor behind the work described in this paper.

The advantage of a parametric curve is especially evident when the scanned object has a simple geometry that needs to be precisely reconstructed. Take for example a man-made rectangular object where the aim is to measure a small deformation in shape. To precisely reconstruct such an object using level-sets, the resolution of the underlying grid needs to be very high (resulting in lots of unknowns). Furthermore, the solution will be influence by the alignment (angle) between the object and the grid. Using a parametric curve, we can precisely represent the object with a small number of points precisely delineating the boundary.

Several further advantages of parametric curve have not yet been exploited in our work. These include the possibility for adaptive resolution, such that the number of points changes depending on the curve's appearance. Moreover, the parametric curve makes it straightforward to compute geometric statistic such as boundary length, area, or curvature. Incorporating shape priors is also easier when using a parametric curve. Finally, the segmentation represented via the parametric curve can be directly used for further data processing using, e.g., finite-element methods.

We expect the full value of our approach to be evident when it is implemented with an topology-adaptive parametric curve, e.g., from [27]. That is outside the scope of this paper – here we only consider a single closed parametric curve. The contribution of this work is a rigorous mathematical derivation of a deformation model for any type of parametric curve. We also suggest an efficient implementation of the associated algorithms for tomographic segmentation. Furthermore, we test the performance of our approach in a controlled setting, and compare our results with state-of-the-art methods.

The method presented here is, in itself, relevant for certain problems where a single closed curve suffices. This situation is seen in electron tomography where scanned samples typically have a shape that can be

easily represented by one curve, giving rise to use of geometric reconstruction methods [30]. Furthermore, reconstruction methods for electron tomography often need to deal with high noise levels and projections taken from a limited angular range – conditions where our method yields promising results.

## 2. Segmentation Algorithm Using a Parametric Curve

In general, a deformable curve is evolved from an initial stage such that it minimizes the segmentation energy. Therefore, the essence of the approach using deformable curves lies in formulating a suitable segmentation energy and deriving the corresponding deformation force.

The contributions to the segmentation energy are divided into *external* and *internal* energies, leading to *external* and *internal* deformation force, respectively. Internal force depends only on the curve; a common choice is to minimize the length and/or bending of the curve, making the method robust to noise. External force depends on the data; in this application it moves the curve to minimize a data-fitting term involving the projection data. In the following description of the method, the focus is on the external contribution, but bear in mind that this is supplemented by the internal energy.

### *2.1. Image Segmentation*

Before describing our tomographic segmentation we present a deformable model for image segmentation using a piecewise constant Mumford-Shah potential. Deformable force for this model is derived in [5] with a focus on level-set representation of the curve. Here we derive identical force, but for a parametric curve representation.

We consider a case where the task is to separate the foreground from the background, and this is achieved by evolving a simple closed curve which partitions the image into the *inside* and *outside*. The model we are to derive applies to arbitrary two-phase topology, but we describe the foreground-background situation for the simplicity of the presentation.

The input for the method is a gray-scale image $g$ defined on an image domain $\Omega$. The deformable curve is represented by a sequence of points $\mathbf{c}_n \in \Omega$, $n = 1, \ldots, N$.

For a given curve $\mathbf{c}$ and two intensity values $m_{\text{in}}$ and $m_{\text{out}}$ the piecewise constant Mumford-Shah external segmentation energy is defined as

$$E_{\text{ext}}(\mathbf{c}, m_{\text{in}}, m_{\text{out}}) = \int_{\Omega} (g - m)^2 \mathrm{d}\mathbf{x}\,, \tag{1}$$

where $m$ is the image which takes a value $m_{\text{in}}$ inside the curve $\mathbf{c}$ and $m_{\text{out}}$ outside.

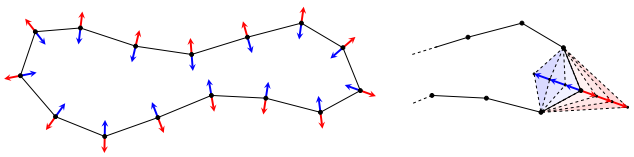

**Figure 3.** Normal displacement of the parametric curve. *Left:* a curve deformation when each point is displaced by 1 (red) or $-1$ (blue) in the normal direction. *Right:* the area affected by displacing one vertex is a linear function of the displacement.

To minimize $E_{\text{ext}}$ we adopt the alternating optimization scheme from [5]. In each iteration of this algorithm we first compute optimal $m_{\text{in}}$ and $m_{\text{out}}$ for a fixed $\mathbf{c}$, and we then evolve $\mathbf{c}$ to minimize the energy for a given $m_{\text{in}}$ and $m_{\text{out}}$.

The first step is straightforward. For a fixed $\mathbf{c}$ the value of $m_{\text{in}}$ affects only integration inside the curve. We have

$$\frac{\mathrm{d}E_{\text{ext}}}{\mathrm{d}m_{\text{in}}} = 2 \int_{\Omega_{\text{in}}} (g - m_{\text{in}}) \mathrm{d}\mathbf{x}\,. \tag{2}$$

Setting this to zero we obtain an optimal value

$$m_{\text{in}} = \frac{\int_{\Omega_{\text{in}}} g \mathrm{d}\mathbf{x}}{\int_{\Omega_{\text{in}}} \mathrm{d}\mathbf{x}}\,. \tag{3}$$

Similarly, $m_{\text{out}}$ is an average of the outside region $\Omega_{\text{out}}$.

For the second step we define the curve evolution in terms of point displacement in the direction of the outward pointing normal, see Figure 3 *left*. Having fixed $m_{\text{in}}$ and $m_{\text{out}}$ we consider slightly displacing one point of the curve

$$\mathbf{c}_n^{\text{new}} = \mathbf{c}_n + \delta_n \mathbf{n}_n\,, \tag{4}$$

where $\delta_n$ is the signed displacement and $\mathbf{n}_n$ is an outward pointing normal vector at $\mathbf{c}_n$.

If $\delta_n > 0$ then the area that correspond to the change in the curve lies outside the curve before it was changed, and inside the curve after it was changed. The opposite is, of course, valid for a negative displacement. See Figure 3 *right* for an illustration. This area is proportional to $\delta_n$ and the coefficient of proportionality depends on the curve parametrization (length of the segments) and the curvature. For simplicity, we assume that the curve points $\mathbf{c}_n$ are placed equidistantly with curve segments of unit length, and that the curve is smooth, so we can omit the coefficient of proportionality. We also assume a small displacement, such that the image intensity in the affected area can be assumed constant.

Let $g_n = g(\mathbf{c}_n)$ denote the image intensity at the point $\mathbf{c}_n$. The energy, as a function of the displacement, is now linear in $\delta_n$:

$$E_{\text{ext}}(\delta_n) = E_0 + (g_n - m_{\text{in}})^2 \delta_n - (g_n - m_{\text{out}})^2 \delta_n\,, \tag{5}$$

with the constant $E_0$ being the external energy before the displacement of $\mathbf{c}_n$. We arrive at

$$\frac{\mathrm{d}E_{\text{ext}}}{\mathrm{d}\delta_n} = (m_{\text{out}} - m_{\text{in}})(2g_n - m_{\text{in}} - m_{\text{out}})\,. \tag{6}$$

Minimizing the energy by moving in the direction of the negative gradient leads to the curve evolution given by

$$\mathbf{c}_n^{\text{new}} = \mathbf{c}_n + \tau(m_{\text{in}} - m_{\text{out}})(2g_n - m_{\text{in}} - m_{\text{out}})\,\mathbf{n}_n\,, \tag{7}$$

where $\tau$ is a user-defined length of the update step.

Displacement of all curve points is computed in parallel and points are displaced at once, followed by a new computation of $m_{\text{in}}$ and $m_{\text{out}}$. This two-step process is repeated until the change in the curve is below a user-defined tolerance.

### *2.2. Tomographic Segmentation*

For tomographic segmentation, we again consider a simple closed curve $\mathbf{c}$, partitioning the reconstruction domain into *inside* and *outside*. Furthermore, we assume that the outside region (the background) has attenuation zero, so we only consider the attenuation of the inside region (the object) which we denote $\mu$. We later show that our modes generalizes to multiple objects of different attenuation, including the case where attenuation of the background is nonzero.

The input to the tomographic segmentation is a discrete sinogram $s_{kj}$ for $j = 1,\ldots,J$ and $k = 1,\ldots,K$. Index $j$ denotes the $j$th detector pixel, while index $k$ corresponds to the set of projection angles $\Theta = \{\theta_k, k = 1,\ldots,K\}$. Again, the discrete simple closed curve is represented by the points $\mathbf{c}_n$, which now are in a reconstruction domain.

For a given attenuation value $\mu$ and a curve $\mathbf{c}$ we define external segmentation energy as the squared residual:

$$E_{\text{ext}}(\mathbf{c},\mu) = \sum_{j=1}^{J}\sum_{k=1}^{K}(s_{kj} - p_{kj})^2\,, \tag{8}$$

where $p_{kj}$ is the discrete predicted sinogram, i.e., a projection of the object of intensity $\mu$ delineated by the curve $\mathbf{c}$.

For our minimization algorithm we describe three contributions:

(i) an efficient algorithm for the forward model, i.e., the computation of $p_{kj}$ given $\mathbf{c}$ and $\mu$,

(ii) the computation of optimal attenuation $\mu$ given $\mathbf{c}$ and $s_{kj}$, and

(iii) the deformation of the curve $\mathbf{c}$ given $\mu$ and $s_{kj}$.

As in image segmentation, the two-step process involves alternating between updating $\mu$ and deforming the curve $\mathbf{c}$, while those two steps rely on computation of $p_{kj}$.

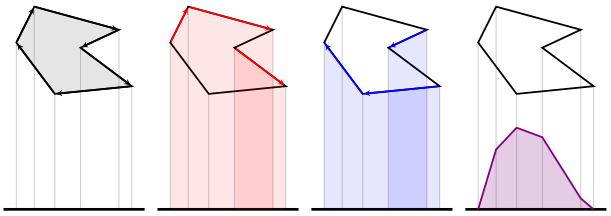

**Figure 4.** Illustration of the forward model for one projection angle. *Left.* The gray object which is to be projected at the detector. Arrows show the orientation of curve segments. *Two illustrations in the middle.* The red and blue indicate the positive and the negative direction of the curve segments when projected to the detector, which defines the sign of corresponding contributions. Note that the darker shade indicates a part of the detector receiving two positive and two negative contributions. *Right.* The resulting projection.

*2.2.1. Efficient Forward Model Computation.* The forward model is concerned with producing projection data given the object represented by the parametric curve $\mathbf{c}$ and the attenuation $\mu$. At first we consider one projection angle, and an object of an attenuation $\hat{\mu} = 1$.

Similar to computing the area enclosed by a curve using Green's theorem, we compute the contribution to all detector pixels by following a curve around the object and considering the distance between the curve and the detector. For our discrete curve, the ordering of curve points $\mathbf{c}_i$ implies an orientation of each curve segment $\overline{\mathbf{c}_n\mathbf{c}_{n+1}}$, for example such that the object is always on the right when travelling along the curve, see Figure 4. The projection of each curve segment to the detector results in either a positive or a negative direction along the detector. This determines whether $\overline{\mathbf{c}_n\mathbf{c}_{n+1}}$ contributes positively or negatively to the affected detector pixels. Following the curve around the object and summing the signed contributions from all segments yields the projection values, and is valid for arbitrary object shape. Contributions from a curve segment to the detector pixels may be computed as distances or areas.

A full projection is obtained by rotating the curve points for other angles. For attenuation different than $\hat{\mu} = 1$ we multiply the result by $\mu$. Pseudocode for the forward model can be found in Algorithm 1, where we with $\ell(n,k)$ denote the detector pixel where the curve point $\mathbf{c}_n$ projects for angle $\theta_k$. Care has to be taken to ensure that $\overline{\mathbf{c}_n\mathbf{c}_{n+1}}$ contributes correctly to $\ell(n,k)$ and $\ell(n+1,k)$, where end-points of the curve segment project.

*2.2.2. Attenuation Update.* Computing the optimal attenuation given a fixed curve is the first step in the update and it is illustrated in Figure 5 *left*. We denote with $\hat{p}_{kj}$ the projection of the curve $\mathbf{c}$ for object attenuation $\hat{\mu} = 1$. Our aim is to find $\mu$ which

**Algorithm 1** Forward model

```
c_{N+1} = c_1                                   ▷ close the curve
initialize s_{kj} with zeros                    ▷ all angles and pixels
for all k = 1, ..., K do                        ▷ angles
  for all i = 1, ..., N do                      ▷ curve segments
    compute ℓ(n, k)                             ▷ projection of c_n for θ_k
    compute ℓ(n + 1, k)                         ▷ projection of c_{n+1} for θ_k
    determine sign                              ▷ compare ℓ(n, k) and ℓ(n + 1, k)
    for all ℓ between ℓ(n, k) and ℓ(n + 1, k) do
      update s_{kl}                             ▷ add signed contribution of c_n c_{n+1}
    end for
  end for
end for
```

minimizes the segmentation energy in Eq. (8). The energy, as a function of attenuation, is

$$E_{\text{ext}}(\mu) = \sum_{j=1}^{J}\sum_{k=1}^{K}(s_{kj} - \mu\hat{p}_{kj})^2\,, \tag{9}$$

and hence

$$\frac{\mathrm{d}E_{\text{ext}}}{\mathrm{d}\mu} = 2\sum_{j=1}^{J}\sum_{k=1}^{K}(\mu\hat{p}_{kj} - s_{kj})\hat{p}_{kj}\,. \tag{10}$$

Setting this to zero we obtain an attenuation which minimizes the energy as

$$\mu = \frac{\sum_{j=1}^{J}\sum_{k=1}^{K} s_{kj}\,\hat{p}_{kj}}{\sum_{j=1}^{J}\sum_{k=1}^{K}(\hat{p}_{kj})^2}\,. \tag{11}$$

This attenuation corresponds to a least squares fit of $\mu\hat{p}_{kj}$ to $s_{kj}$.

After updating the attenuation, the projection is given by

$$p_{kj} = \mu\hat{p}_{kj}\,. \tag{12}$$

*2.2.3. Curve Deformation.* Just as in Eq. (4) we consider a small displacement in one point of the curve, which in case of a positive displacement causes a small area to leave the background and enter the object. Using the same argument as before, we conclude that this area is proportional to the step length $\delta_n$.

We assume that the displacement is small, so for every projection angle $\theta_k$ the change in the projection $p_{kj}$ is concentrated in the detector pixel where $\mathbf{c}_n$ is projected. We denote the value of the sinogram in this point as $s_{k\ell(n,k)}$, where $\ell(n,k)$ is still the detector pixel where the curve point $\mathbf{c}_n$ projects for angle $\theta_k$. The corresponding value in the predicted sinogram is $p_{k\ell(n,k)}$.

The energy from Eq. (8), as the function of displacement, is quadratic:

$$E_{\text{ext}}(\delta_n) = E_0 + \sum_{k=1}^{K}\big[(s_{k\ell(n,k)} - p_{k\ell(n,k)} + \mu\delta_n)^2 - (s_{k\ell(n,k)} - p_{k\ell(n,k)})^2\big]\,, \tag{13}$$

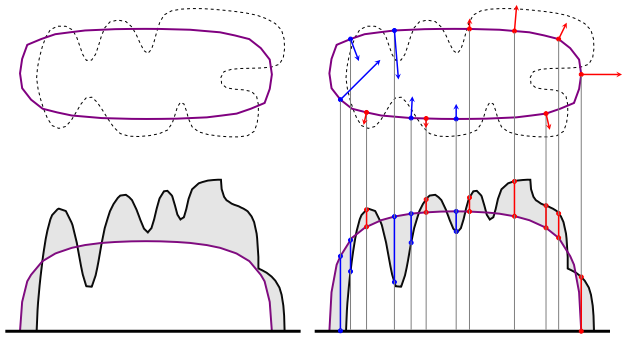

**Figure 5.** Illustration of the update. *Left.* An unknown object (dashed line) has projection $s$ (black). A curve (purple) very roughly approximates the object and assuming $\hat{\mu} = 1$ has projection $\hat{p}$ (also in purple). Residual $s - \hat{p}$ is shown as the gray area. *Right.* Updating the attenuation results in a projection $p = \mu\hat{p}$ (purple) which minimizes the squared residual. For the curve deformation step we evaluate the residual at the detector pixels where the curve points are projected, here drawn for a subset of points (positive in red, negative in blue). Residuals (divided by $\mu$) give the direction and the size of the normal displacement of the points on the curve (outwards in red, inwards in blue).

where $E_0$ is the energy before displacing the point $\mathbf{c}_n$. We have

$$\frac{\mathrm{d}E_{\text{ext}}}{\mathrm{d}\delta_n} = 2\mu\sum_{k=1}^{K}\left(s_{k\ell(n,k)} - p_{k\ell(n,k)} + \mu\delta_n\right)\,. \tag{14}$$

Setting this to zero we find the displacement which minimizes the external energy as

$$\delta_n = \frac{1}{\mu}\frac{1}{K}\sum_{k=1}^{K}(p_{k\ell(n,k)} - s_{k\ell(n,k)})\,. \tag{15}$$

And curve update is

$$\mathbf{c}_n^{\text{new}} = \mathbf{c}_n + \delta_n\mathbf{n}_n\,. \tag{16}$$

The above principle is illustrated for a single projection angle in Figure 5 *right*. For each curve point $\mathbf{c}_n$, the sinogram residual $p_{k\ell(n,k)} - s_{l\ell(n,k)}$ is evaluated at the detector pixel $\ell(n,k)$ where the curve point is projected. The value of the residual contributes to the normal displacement of the curve.

Note in Figure 5 that some displacements may be resolved successfully from only one projection angle, while resolving others requires information from other angles. This is the role of averaging the contributions across the angles. Also note that the summation in (16) corresponds to a line integral in the residual along the sine curves defined by projection of $\mathbf{c}_n$ on the detector.

*2.2.4. The Resulting Algorithm.* The steps in our tomographic segmentation are summarized in Algorithm 2. The outcome of the algorithm is a binary segmentation (and simplified reconstruction) in the reconstruction domain – a curve $\mathbf{c}$, which delineates

a region in the reconstruction domain which has an attenuation $\mu$. Similar to the Chan-Vese algorithm [5] and related methods, our algorithm converges to a local minimum of the total energy, consisting of the external energy $E_{\text{ext}}$ plus the regularization term.

**Algorithm 2** Tomographic segmentation

| | |
|---|---|
| initiate curve $\mathbf{c}$ | ▷ see Sec. 2.4 |
| **while** not converged **do** | ▷ see Sec. 2.4 |
| compute $\hat{p}_{kj}$ for $\mathbf{c}$ and $\hat{\mu} = 1$ | ▷ Algorithm 1 |
| compute $\mu$ for $\hat{p}_{kj}$ and $s_{kj}$ | ▷ Eq. (11) |
| compute $p_{kj}$ | ▷ Eq. (12) |
| compute $\delta_n$ for $p_{kj}$ and $s_{kj}$ | ▷ Eq. (15) |
| deform the curve | ▷ Eq. (16) |
| regularize the curve | ▷ see Sec. 2.4 |
| **end while** | |

### *2.3. Complexity Analysis*

To support the claim that our algorithm is run-time and memory efficient, we give a complexity analysis for one iteration. In this context, the size of the problem is given by the number of projection angles $K$, the number of detector pixels $J$, and the number of curve points $N$. We assume that points are equidistantly placed around the curve, and that the length of the curve does not depend on the size of the problem.

The most complicated part of our algorithm is the forward model, Algorithm 1, which is implemented as three nested loops. The first and the second loop run over projection angles ($K$ times) and over curve segments ($N$ times). The third loop runs over all detector pixels that are affected by the curve segment. The number of affected pixels is proportional to $J$ and inversely proportional to $N$, since larger $N$ implies smaller line segments. Due to overhead in the third loop, it iterates $\mathcal{O}(J/N + 1)$ times. Putting this together we find that the forward projection takes $\mathcal{O}\left(K(J + N)\right)$ time. Regarding the remaining steps in the algorithm, the attenuation estimation is of order $\mathcal{O}(KJ)$, while computing the curve deformation takes $\mathcal{O}(KN)$ time. As for the memory requirements of the whole framework, only the current curve and the current sinogram need to be stored, and that requires $\mathcal{O}(KJ + N)$ memory.

Along the same lines, let us mention the possibilities of additionally improving the efficiently of the algorithm. One strategy involves computing values of the forward projection only where we need them – along the sine curves in the sinogram $p$ defined by the projection of curve points. These are sufficient for computing deformation forces, and also for estimation of the attenuation coefficient which is a strongly overdetermined system and will not be affected by removing some of the fitting terms. With this strategy the run-time is independent of the number of detector pixels, but it requires sorting the projected points on the detector, resulting in an algorithm of the order $\mathcal{O}\left(KN \log N\right)$.

Another strategy worth mentioning is parallelization. Operations on projections angles and curve points are fully independent, and this allows for an efficient parallel implementation.

### *2.4. Implementation*

For our current implementation, we made a number of implementation choices, partly to alleviate the limitations of our simple curve representation. First, we keep the number of curve points constant during evolution. The curve is initialized as a circle in the center of the reconstruction domain. External and internal forces deform the curve sequentially. For better stability, when deforming according to Eq. (16) we introduce a user-defined length of the update step $\tau < 1$ and move the point $\mathbf{c}_n$ for $\tau\delta_n$.

For regularization, we chose an implementation which uses finite differences [3, 31]. In this formulation, each curve point is being smoothed trough contributions from its neighbours. Those are defined by

$$\begin{aligned}\Delta\mathbf{c}_n = \alpha(\mathbf{c}_{n-1} - 2\mathbf{c}_n + \mathbf{c}_{n+1}) + \beta(-\mathbf{c}_{n-2} + 4\mathbf{c}_{n-1} \\ - 6\mathbf{c}_n + 4\mathbf{c}_{n+1} - \mathbf{c}_{n+2})\,,\end{aligned} \tag{17}$$

where $\alpha$ controls an elasticity term (minimizes curve length) and $\beta$ controls an rigidity term (minimizes curve bending). A backward Euler step is used for better stability, particularly important under strong regularization, and the update is

$$\mathbf{c}_n^{\text{new}} - \Delta\mathbf{c}_n^{\text{new}} = \mathbf{c}_n\,. \tag{18}$$

This can be calculated efficiently, see [3] or [31] for details.

Our algorithm can be sensitive to irregular length of curve segments. We therefore complete each iteration of the curve deformation with a step which distributes points equidistantly along the curve.

We foresee using total curve deformation for each iteration (including both the fitting update and the regularization step) measured as

$$f = \sum_{n=1}^{N} \|\mathbf{c}_n^{\text{new}} - \mathbf{c}_n\|_2 \tag{19}$$

as a stopping criterium. However, in our current implementation we stop the evolution after a fixed number of iterations.

### *2.5. Additional Remarks*

Our current implementation considers parallel-beam X-ray tomography, and our discretization uses the so-called line model. Hence, each detector pixel is a single

point and we consider the line from the source to the pixel that intersects the particular curve segment being treated in the algorithm's inner loop. This can be easily extended to a strip model where the detector pixel has a finite width – and we would then consider a the strip associated with the detector and the curve segment.

Likewise, a fan-beam model can be implemented. For the fan-beam line model, we would again consider the line from the source to the point pixel, while for the strip model we would consider the area of the triangle defined by the source and the pixel.

Notice that our model easily handles cases when the object is not seen on the detector for some angles. When computing the displacements in Eq. (16), if some of the curve points project outside the detector then the corresponding angles should be ignored, and averaging should be performed only over the present data. This leads to correct deformation of the curve, despite the missing parts. In a similar fashion we can handle cases where the sinogram is partially corrupted, for example by faulty pixels.

Our curve deformation model may be generalized to address a multi-phase reconstruction problem. The generalization would require a number of critical implementation choices, and therefore we here only provide a rough outline of such an algorithm where, for clarity, we drop the subscripts for the sinograms. Consider a partition of the reconstruction domain into $M$ regions, defined by points connected with line segments. Regions have attenuations $\mu_m$, $m = 1, \ldots, M$. For the forward model, each region independently produces $\hat{p}_m$ and the projection is

$$p = \sum_{m=1}^{M} \mu_m \hat{p}_m \,. \tag{20}$$

Computing optimal attenuation corresponds to finding least squares solution for a system

$$\sum_{m=1}^{M} \mu_m \hat{p} = s \tag{21}$$

where every sinogram pixel contributes with one equation. Finally, for curve deformation consider a point on a curve between regions $m$ and $m'$. Assuming that the curve normal points to region $m$, the displacement in Eq. (15) should be modified such that $\mu$ is replaced by $\mu_{m'} - \mu_m$. A simple version of a multi-phase segmentation is the case where the background attenuation is not zero, but is calculated from the data.

## 3. Results

In the first batch of examples in Figures 6–12 the ground truth object is represented using a curve, and we use our own forward model (Algorithm 1) to also compute the reconstruction, therefore performing so-called inverse crime. The focus of these first experiments is on the performance of our method in the presence of noise and with a limited number of projections. We emphasize that we use two different forward models in the subsequent experiments.

Figure 6 gives a systematic overview of the experimental setup used in these experiments. With $\tilde{o}$ we denote the object in the reconstruction domain, where tilde indicates that this is unknown in an actual inverse problem. In this example the object is a polygon of a constant attenuation $\tilde{\mu}$. The corresponding sinogram, here constructed using our forward model, is denoted $\tilde{s}$ (we again drop the subscripts for clarity). We highlight a few of polygon vertices and the corresponding sine curves in $\tilde{s}$.

As the input to the reconstruction problem we use a noise-corrupted sinogram

$$s = \tilde{s} + e \,, \tag{22}$$

where $e$ is zero-mean Gaussian noise, and the relative noise level is set by a parameter

$$\eta = \sqrt{\frac{\sum s^2}{\sum e^2}} \,. \tag{23}$$

In Figure 6, for an illustrative example, we use a relatively high noise level, combined with a strong regularization and a small length of the update step.

The curve is initialized as a circle $\mathbf{c}^{\text{init}}$, and again we highlight two of its points and the corresponding sine curves in corresponding initial forward projection $\hat{p}^{\text{init}}$ calculated for $\hat{\mu} = 1$. After calculating the attenuation coefficient $\mu^{\text{init}}$ we have the initial predicted sinogram $p^{\text{init}}$ (not shown) and the initial reconstruction $r^{\text{init}}$, which is an object of attenuation $\mu^{\text{init}}$ delineated by $\mathbf{c}^{\text{init}}$. We show the initial residual $s - p^{\text{init}}$, and also here we highlight the projections of the two curve points. Notice that one sine curve lies in predominately positive areas, while another lies in predominately negative areas of the residual. This leads to a deformation where the first point moves outwards, and the second moves inwards, as shown in the illustration of the curve evolution.

For the outcome after the last iteration we show different results: the resulting curve $\mathbf{c}^{\text{end}}$, the reconstruction $r^{\text{end}}$, the corresponding sinogram $p^{\text{end}}$, and residual sinogram $s - p^{\text{end}}$. Since we have access to noise-free data, we can also show the errors $\tilde{s} - p^{\text{end}}$ and $\tilde{o} - r^{\text{end}}$ in both the projection (sinogram) domain and the reconstruction domain.

Figure 6 also includes graphs showing convergence of different quantities as a function of the iterations: the estimated attenuation coefficient, the residuals and errors, and the curve deformation. For total curve deformation measured by Eq. (19), we show

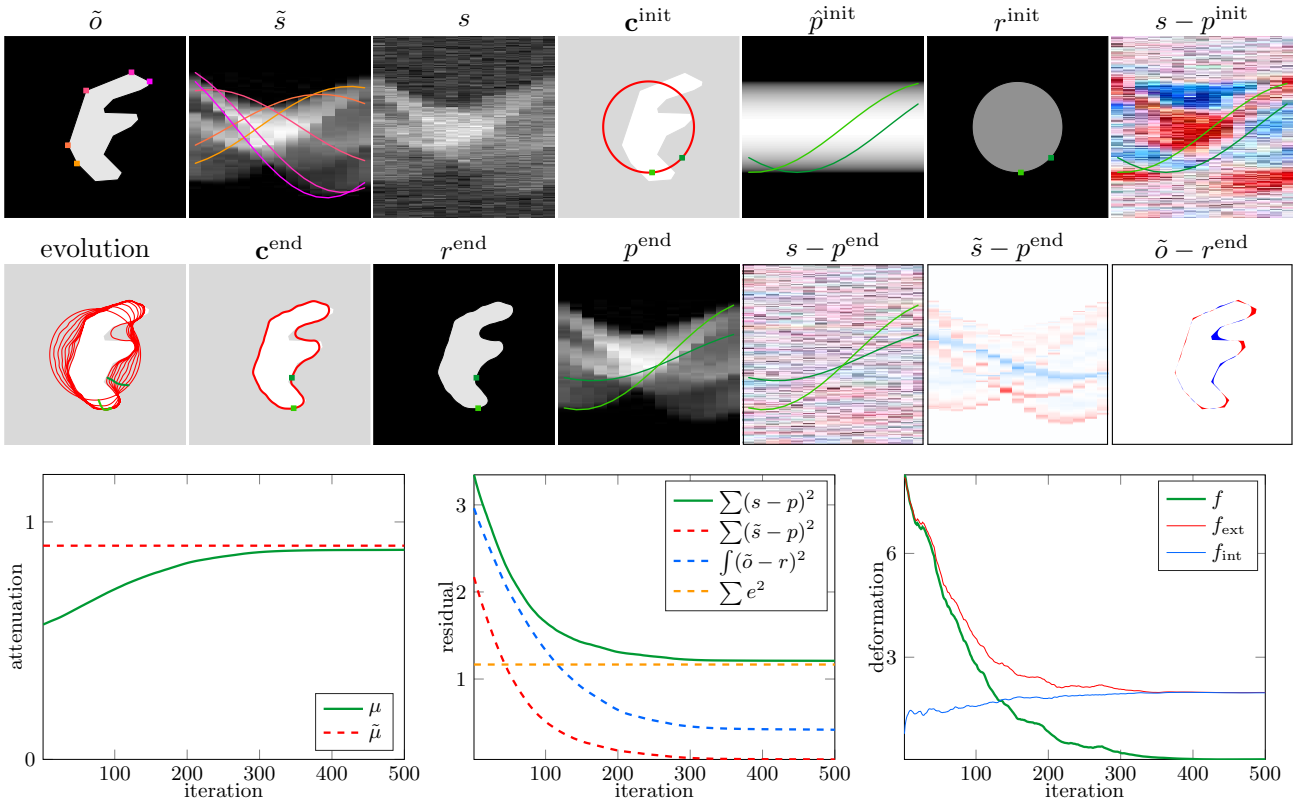


**Figure 6.** Our experimental setup. For this experiment we used $\eta = 0.3$, $\alpha = \beta = 1$ and $\tau = 0.2$. *Top row:* test object $\tilde{o}$, noise-free sinogram $\tilde{s}$, noisy sinogram $s$, curve initialization $\mathbf{c}^{\text{init}}$, initial projection (before adjusting attenuation) $\hat{p}^{\text{init}}$, initial reconstruction $r^{\text{init}}$, and initial residual $s - p^{\text{init}}$. *Middle row:* Curve evolution shows every 25th iteration, resulting curve $\mathbf{c}^{\text{end}}$, resulting reconstruction $r^{\text{end}}$, resulting predicted sinogram $p^{\text{end}}$, resulting residual $s - p^{\text{end}}$, resulting error in projection domain $\tilde{s} - p^{\text{end}}$, and resulting error in object domain $\tilde{o} - r^{\text{end}}$. When showing the curve, we always draw it over the faded image of the test object. For showing residual end errors we use blue color for negative values, white for zero, and red for positive values. *Bottom row:* Attenuation coefficient, residual and errors, and curve deformation over iterations.

also the two contributions: deformation $f_{\text{ext}}$ after the external (data-driven) update defined by Eq. (16), and deformation $f_{\text{int}}$ after the internal (curve-driven) update from Eq. (18). Note that in our implementation $f_{\text{ext}} = \tau \sum |\delta_i|$.

We see that the estimated attenuation converges to the (unknown) true value. The total curve deformation converges to zero when the internal and external contributions balance each other. The norm of the residual converges to a constant value, and as we have access to true values we can confirm this to be slightly larger than the norm of the noise.

Figure 7 introduces our test object depicting a dinosaur. For this evolution we use the default setting listed in Table 1. Unless stated otherwise, all conducted experiments use the default setting. In particular, evolution parameters are constant for all experiments, and curve regularization parameters change only in an experiment with focus on regularization. Furthermore, unless stated differently, projection angles are equidistantly sampled from 0 to $\pi$, the detector width equals the width of the reconstruction domain

| | symbol | value | |
|---|---|---|---|
| Projection | $\eta$ | 0.1 | relative noise level |
| | $K$ | 15 | no. projection angles |
| | $J$ | 200 | no. detector pixels |
| Curve | $N$ | 500 | no. curve points |
| | $\alpha$ | 0.01 | curve elasticity |
| | $\beta$ | 0.01 | curve rigidity |
| Evolution | $\tau$ | 0.05 | update step length |
| | $T$ | 500 | no. iterations |

**Table 1.** Default settings used for all experiments, unless stated otherwise.

shown in the image, and the curve is initiated as a circle with a diameter which is half of the detector width. In Figure 7 we still show curve evolution, resulting predicted sinogram and residual, but for the remainder of the section we only show $\mathbf{c}^{\text{end}}$ or $r^{\text{end}}$.

Figure 8 demonstrates the performance of the method under increasing noise, where the first is lower,

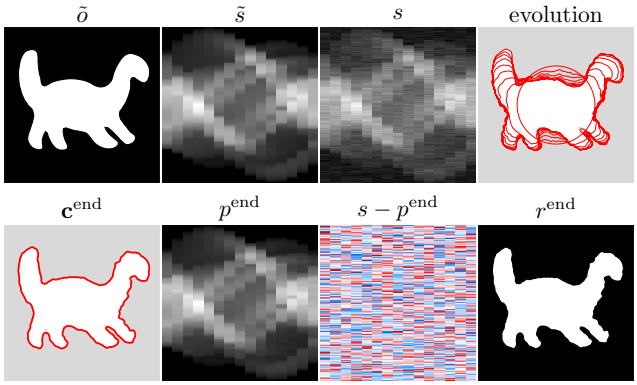


**Figure 7.** An experiment with default settings, see Table 1. *Top row:* a test object $\tilde{o}$, a noise-free sinogram $\tilde{s}$, a noisy sinogram $s$, and curve evolution showing every 25th iteration. *Bottom row:* a resulting curve $\mathbf{c}^{\text{end}}$, a resulting predicted sinogram $p^{\text{end}}$, a resulting residual $s - p^{\text{end}}$, and a resulting reconstruction $r^{\text{end}}$.

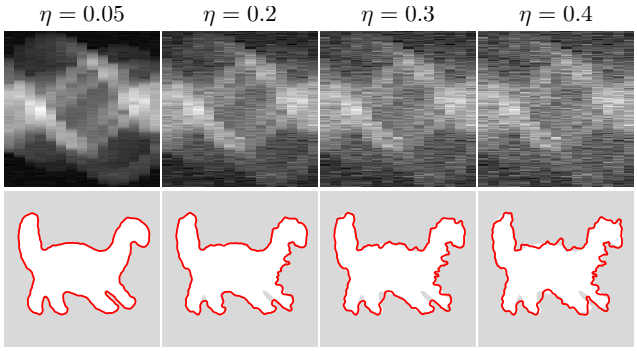


**Figure 8.** An effect of increasing the relative noise level $\eta$. *Top row:* noisy sinograms $s$. *Bottom row:* resulting curves $\mathbf{c}^{\text{end}}$.

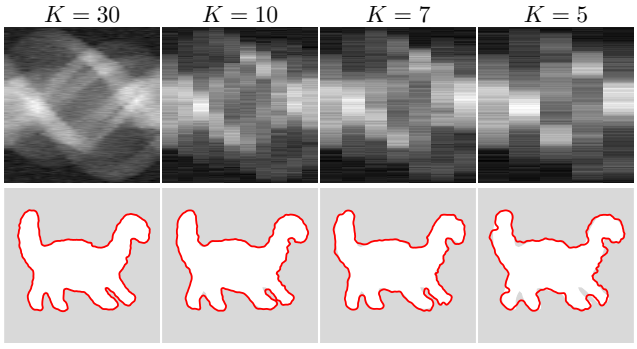


**Figure 9.** An effect of reducing the number of projection angles $K$. *Top row:* noisy sinograms $s$. *Bottom row:* resulting curves $\mathbf{c}^{\text{end}}$.

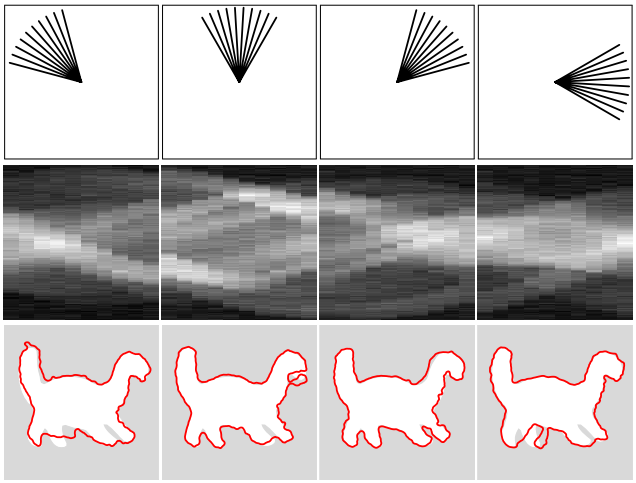

**Figure 10.** An experiment with limited angle reconstruction. We use $K = 10$ projection angles equidistantly sampled from 0 to $\pi/3$ with changing projection direction. *Top row:* directions of the projection angles. *Middle row:* noisy sinograms $s$. *Bottom row:* resulting curves $\mathbf{c}^{\text{end}}$.

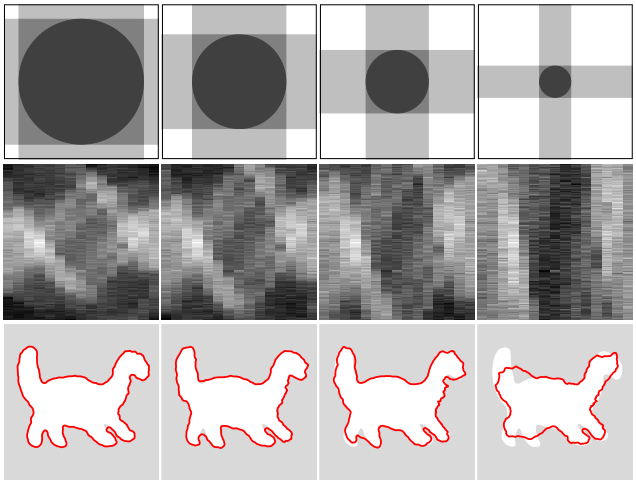

**Figure 11.** An experiment with a detector which is small relative to the size of the object. We kept the number of detector pixels constant, but have reduced the width of the detector to 0.8, 0.6, 0.4 and 0.2 of the original width. *Top row:* illustration indicating the width of the detector for two orthogonal projections, and the circular area in the reconstruction domain which is seen at the detector from all angles. *Middle row:* noisy sinograms $s$. *Bottom row:* resulting curves $\mathbf{c}^{\text{end}}$.

and the last three are higher than the value used in Figure 7. Similarly, Figure 9 shows results when changing the number of projection angles. We also investigated a narrow range of projection angles, and those results are shown in Figure 10. In Figure 11 we change the width of the detector, but we keep the number of pixels constant. In all these experiments we keep regularization parameters constant, we see some over-smoothing in the low-noise and many-angles

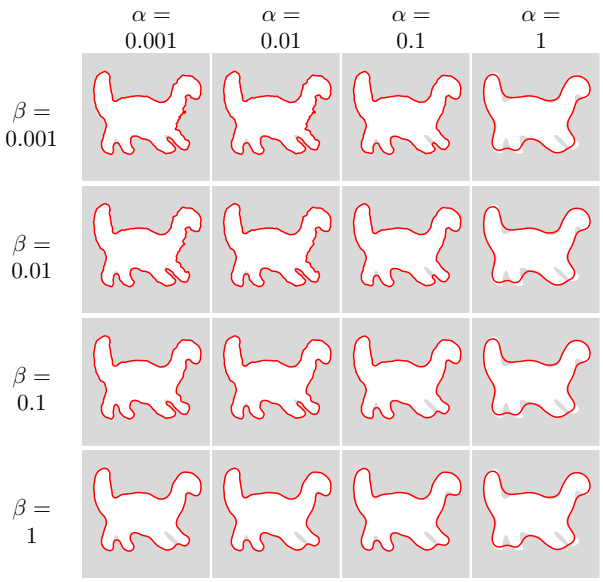


**Figure 12.** An effect of changing curve regularization parameters. The elasticity $\alpha$ changes between columns and rigidity $\beta$ changes between rows.

settings. Adjusting regularization according to the noise level and number of projection angles would lead to better results.

The effect of regularization is evident in Figure 12. These results verify the findings from [32] that increasing $\alpha$ shrinks the curve while $\beta$ smooths the curve without the shrinkage.

The next experiment, in Figure 13, compares our results with the simultaneous algebraic reconstruction technique (SART) [33] and discrete algebraic reconstruction technique (DART) [1]. SART is known for performing well when the projection data is limited, and DART supplements this with an assumption on only a few attenuation levels. We used the implementations provided in the ASTRA toolbox [34, 35]. For this experiment we converted our test object into a $500 \times 500$ pixels binary image. The sinogram $\hat{s}$ was created using a forward model from the ASTRA toolbox, where we used 200 detector pixels as in other experiments. All reconstructions, including ours, have then been tested on the same sinograms. With SART and DART we reconstructed a $200 \times 200$ pixels image. When running SART, we performed 500 iterations with nonnegativity constraints. For DART the pipeline consisted of 100 SART iterations followed by 200 DART iterations with 5 SART sub-steps in each. DART was provided with correct values for the attenuation range between 0 and 1, and the correct threshold value of 0.5. Our algorithm does not rely on that extra input. Compared to SART, our method delineates the objects shape more precisely. This is especially noticeable for the low number of projection angles where SART results display streak artefacts. In comparison with DART, our curve-based representation and regularization is more successful in handling the hight noise levels. The performance of DART would presumably improve had we used the more robust versions SDART [24] and TVR-DART [25] which were not available in the ASTRA toolbox.

In Figure 14 we show four additional objects in form of $200 \times 200$ pixels binary images. In this, and in the remainder of experiments, we used the forward model from AIR Tools [36]. This experiment shows the ability of our curve to capture relatively convoluted outlines, with both rounded and sharp corners.

The experiment in Figure 15 tests the assumption about approximately constant attenuation. The first object displays homogeneous texture at relatively small scale. Thanks to the averaging, we are able to handle this case. The texture of the second object has larger dark areas which affect the result at the boundary of the object. The third object has a blurred outline, which we are capable of capturing. The last examples features non-homogeneous object, which seriously influences the averaging approach of our method captures. The examples in Figure 15 show the global property of our method. Unlike methods that fit the curve to local features as e.g. sharp edges [17, 18], our curve is deformed so that the resulting segmentation of the reconstruction domain fits the data according to the global energy from Eq. (8). We are therefore able to capture homogeneous variations and smooth transitions in attenuation.

Figure 16 shows test cases with multiple objects. It is evident that the curve deformation model also handles this case, and the limitation of our current implementation lies in representing several objects by using a *single closed* curve. A parametric curve with topological adaptivity, which we plan to use in our future work, is expected to alleviate these issues. The last example, shown in Figure 17, shows a case where the topology of the object is simple, but its shape is challenging. The curve was deformed in a convoluted solution, ending in a non-optimal local minimum. Also from this example it is evident that the limitation lies in the representation using a *single closed* curve, not the deformation model. To successfully segment the objects from Figures 16 and 17 our deformation model needs to be combined with a curve representation which is parametric, but can handle topological changes, e.g. a curve from [27].

To confirm our run-time analysis, we tested the performance of our method when the size of the problem increases. We varied three parameters: the number of projection angles, the number of detector pixels, and the number of curve points. We measured the average run-time per iteration over 500

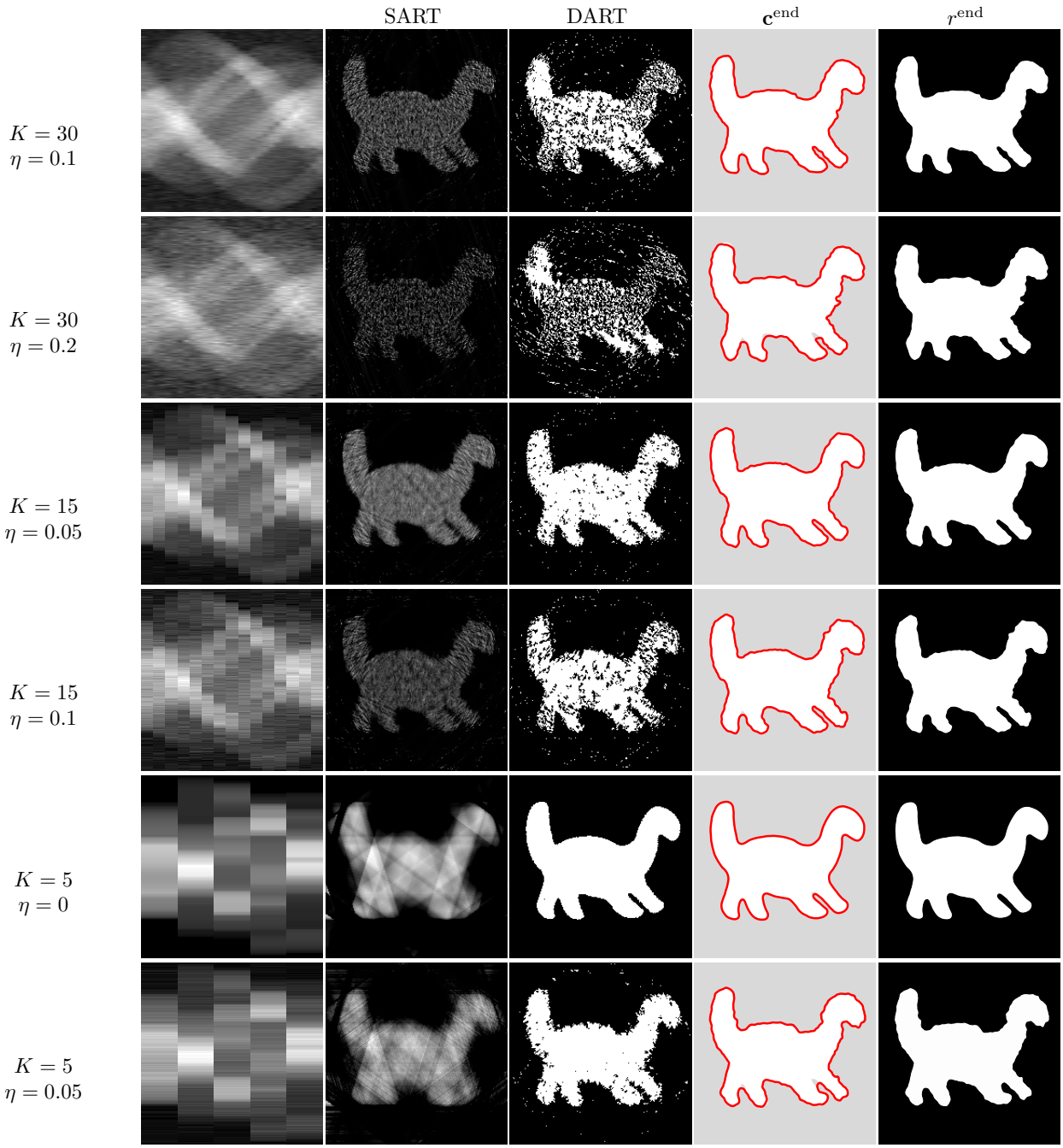


**Figure 13.** Comparison with SART and DART. In each row we use a different setting for a number of projection angles $K$ and relative noise level $\eta$. We show a noise-corrupted sinogram $s$, a SART reconstruction, a DART reconstruction, resulting curve $\mathbf{c}^{\text{end}}$ obtained using our method, and a corresponding resulting reconstruction $r^{\text{end}}$.

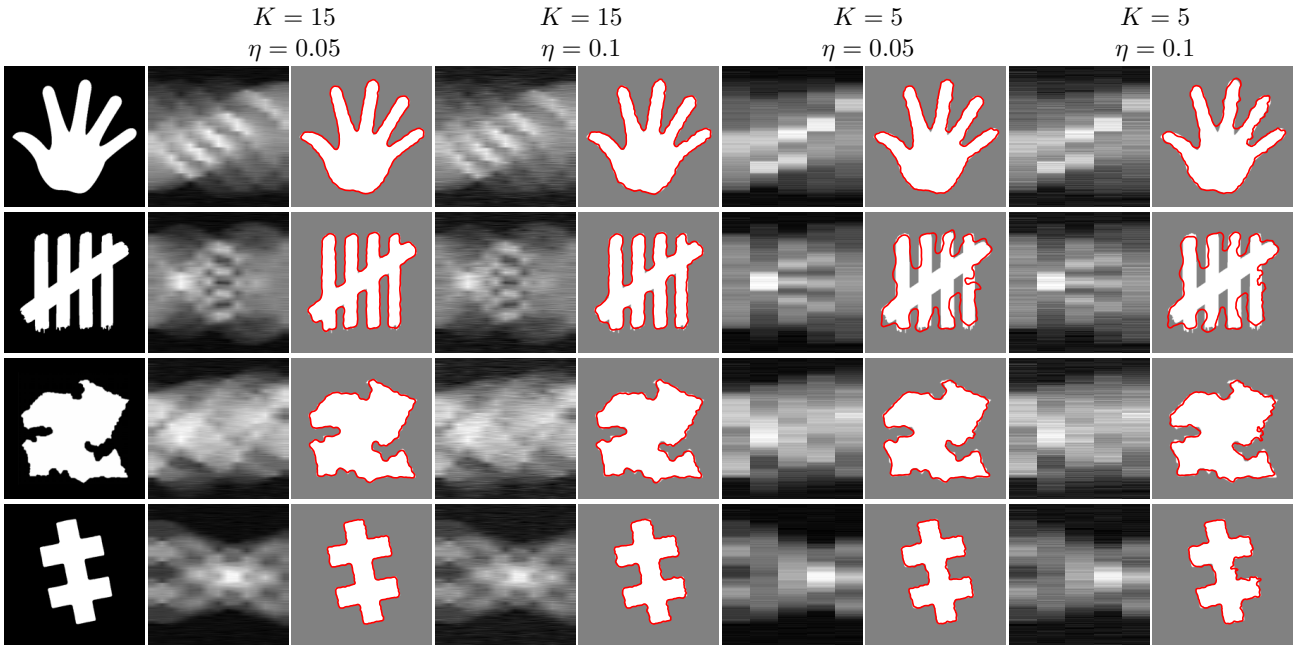


**Figure 14.** Various test objects. In each row we show a test object $\tilde{o}$, a noise-corrupted sinogram $s$, and a resulting curve $\mathbf{c}^{\text{end}}$ for four different settings. Each curve is drawn over the faded image of the test object.

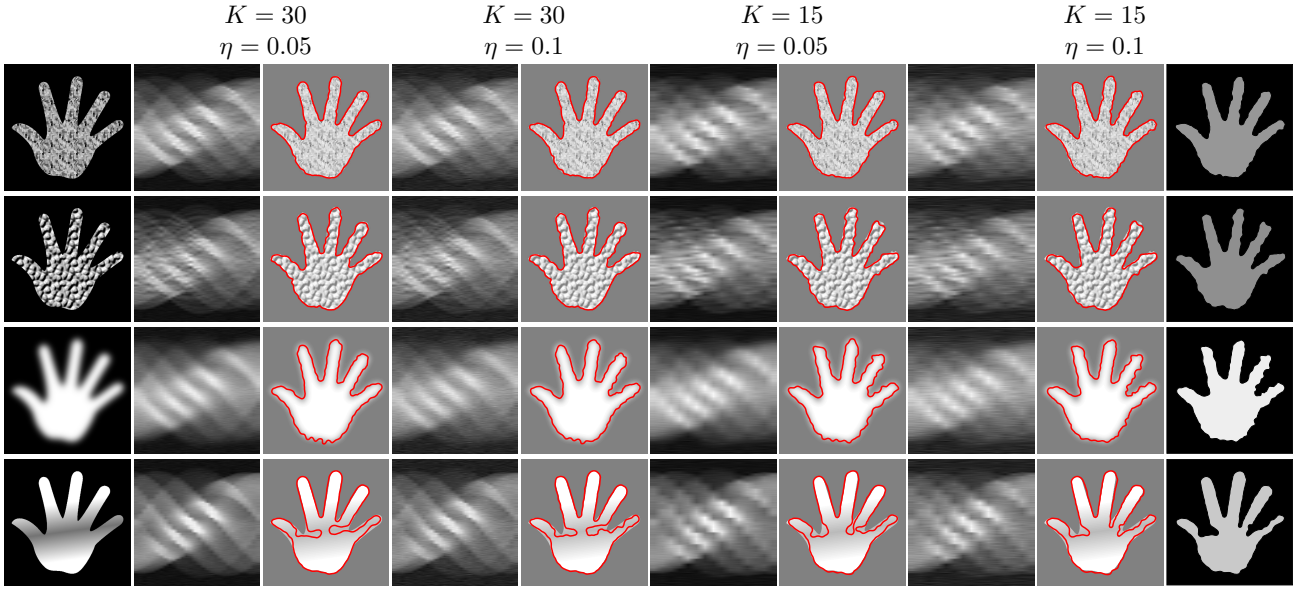


**Figure 15.** Performance when the assumption about approximately constant gray level does not hold. In each row we show a test object $\tilde{o}$, a noise-corrupted sinogram $s$, and a resulting curve $\mathbf{c}^{\text{end}}$ for four different settings. For the most challenging setting, we show also a resulting reconstruction $\mathbf{c}^{\text{end}}$. The estimated attenuation coefficient visible in the rightmost column should be compared with gray level of the test object in the leftmost column.

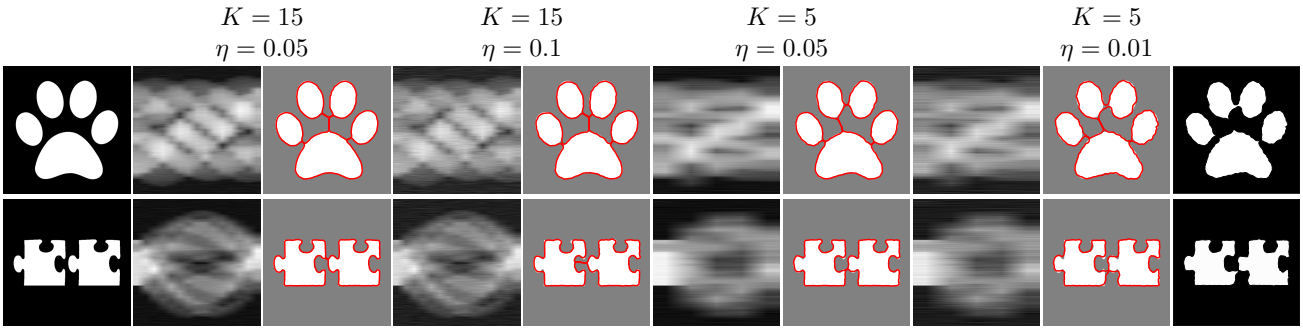


**Figure 16.** Performance when assumption about one object does not hold. In each row we show a test object $\tilde{o}$, a noise-corrupted sinogram $s$, and a resulting curve $\mathbf{c}^{\text{end}}$ for four different settings. For the most challenging setting, we show also a resulting reconstruction $\mathbf{c}^{\text{end}}$.

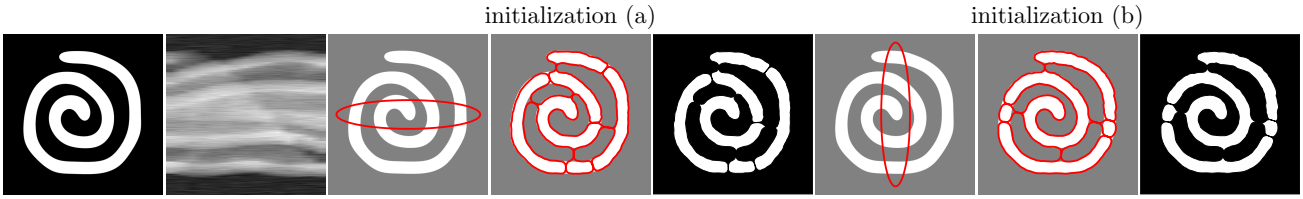


**Figure 17.** Simple topology but challenging shape. The same noise-corrupted sinogram ($K = 15$, $\eta = 0.1$) has been used twice. To obtain two clearly different outcomes for the same reconstruction problem, we perturbed initialization by stretching the circle into an elipse, horizontally for (a) and vertically for (b). For this experiment we used $J = 500$, and otherwise default settings.

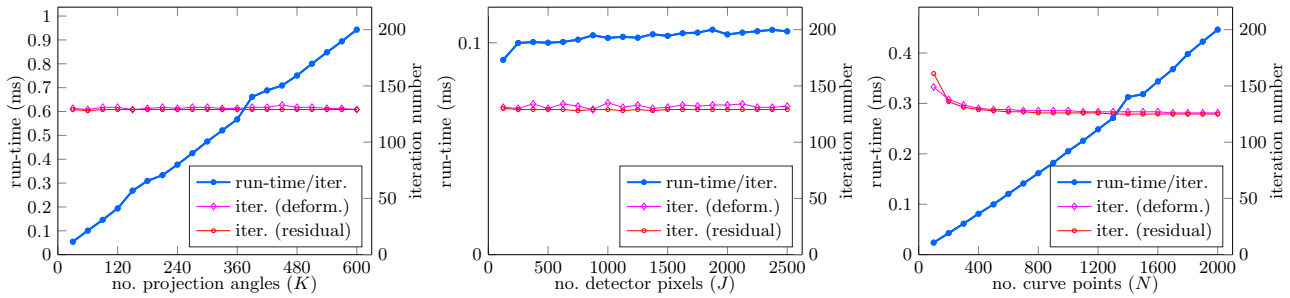


**Figure 18.** Run-time and convergence experiments. The blue line is the average run-time per iteration (in ms), when changing the size of the problem. The red and magenta lines indicate the number of iterations needed until measures used to estimate the convergence drop below a user-defined tolerance. We use two measures: the mean squared residual and the mean curve deformation.

iterations. Furthermore, to estimate the convergence of our method we recorded the iteration number where the mean squared residual drops below a user-defined tolerance level, and an iteration number where the mean curve deformation ($f/N$) drops below a corresponding level. Results of this experiment are shown in Figure 18. The first thing to notice is that the two evaluation criteria convergence agree, and that the number of iterations until convergence seems rather insensitive to the size of the problem. The only systematic trend is seen where an insufficient number of curve points causes a slower convergence (the rightmost plot). As for the run-time per iteration, the results are consistent with our analysis yielding an $\mathcal{O}(K(J+N))$ dependency. Furthermore, the number of detector pixels is the parameter with the weakest influence on run-time, which is only slightly increased when using 2500 detector pixels compared to 200.

Let us add a final remark on the convergence of our method. In our current implementation we evolve the curve with a relatively small length of the update step and we take more than hundred iterations. These small steps ensure that we avoid issues with curve self-intersections, especially under high noise levels. Our initial experiments with larger update steps indicate that convergence may be obtained much faster, in ten or twenty iterations. Also in this respect we expect to benefit from a topology-adaptive representation of

a parametric curve from [27], which automatically resolves selv-intersections.

## 4. Conclusion

Inspired by the use of parametric deformable curves for image segmentation [3], and the models for piecewise constant image segmentation [5], we developed an algorithm that computes a segmented tomographic reconstruction directly from the X-ray projection data. Our algorithm can be considered a hybrid between parametric active contours (such as snakes) and methods based on level-sets that minimize an energy function defined in the projection domain.

By representing the segmentation via a parametric curve we completely avoid the need for a memory-demanding pixel/voxel representation of the object. The key components of our algorithm are efficient computations of the forces that define the deformation of the curve, and efficient computations of the forward projection via a loop over the points that define the curve. Careful simulation results demonstrate the effectiveness and robustness of our methods with respect to noise and situations with few projection data. In addition, we show that the run-time of our algorithm increases linearly with the size of the input sinogram.

The algorithm presented here expands an existing toolbox of the reconstruction methods. It can become a very good solution for reconstruction problems which require a precise geometry of an object consisting of few homogeneous materials.

## Acknowledgements

We thank Michael Andersen for discussions during the initial stages of this project. We also thank two anonymous referees for valuable comments that helped to improve the presentation. The work was funded by Advanced Grant No. 291405 "HD-Tomo" from the European Research Council, and by Grant No. 1305-00032B "CINEMA: the allianCe for ImagiNg of Energy MAterials" from Innovation Fund Denmark.

## References

[1] Kees Joost Batenburg and Jan Sijbers. DART: a practical reconstruction algorithm for discrete tomography. *IEEE Transactions on Image Processing*, 20(9):2542–2553, 2011.

[2] Ross T Whitaker and Vidya Elangovan. A direct approach to estimating surfaces in tomographic data. *Medical Image Analysis*, 6(3):235–249, 2002.

[3] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[4] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[5] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[6] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[7] Vidya Elangovan and Ross T Whitaker. From sinograms to surfaces: A direct approach to the segmentation of tomographic data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 213–223. Springer, 2001.

[8] Jean-Pierre Bruandet, Francoise Peyrin, Jean-Marc Dinten, and Michel Barlaud. 3d tomographic reconstruction of binary images from cone beam projections: A fast level set approach. In *Proceedings IEEE International Symposium on Biomedical Imaging*, pages 677–680. IEEE, 2002.

[9] Eric Debreuve, Michel Barlaud, Gilles Aubert, and Jacques Darcourt. Attenuation map segmentation without reconstruction using a level set method in nuclear medicine imaging. In *International Conference on Image Processing (ICIP)*, volume 1, pages 34–38. IEEE, 1998.

[10] Eric T Chung, Tony F Chan, and Xue-Cheng Tai. Electrical impedance tomography using level set representation and total variational regularization. *Journal of Computational Physics*, 205(1):357–372, 2005.

[11] Ronny Ramlau and Wolfgang Ring. A Mumford–Shah level-set approach for the inversion and segmentation of x-ray tomography data. *Journal of Computational Physics*, 221(2):539–557, 2007.

[12] Christopher V Alvino and Anthony J Yezzi. Tomographic reconstruction of piecewise smooth images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE, 2004.

[13] Sungwon Yoon, Angel R Pineda, and Rebecca Fahrig. Level set reconstruction for sparse angularly sampled data. In *Nuclear Science Symposium Conference Record*, volume 6, pages 3420–3423. IEEE, 2006.

[14] Sungwon Yoon, Angel R Pineda, and Rebecca Fahrig. Simultaneous segmentation and reconstruction: A level set method approach for limited view computed tomography. *Medical Physics*, 37(5):2329–2340, 2010.

[15] Yonggang Shi and W Clem Karl. A multiphase level set method for tomographic reconstruction of dynamic objects. In *Workshop on Statistical Signal Processing*, pages 182–185. IEEE, 2003.

[16] Tony F Chan, Hongwei Li, Marius Lysaker, and Xue-Cheng Tai. Level set method for positron emission tomography. *International Journal of Biomedical Imaging*, 2007, 2007.

[17] Jean-Philippe Thirion. Segmentation of tomographic data without image reconstruction. *IEEE Transactions on Medical Imaging*, 11(1):102–110, 1992.

[18] Steve Sullivan, Alison Noble, and Jean Ponce. On reconstructing curved object boundaries from sparse sets of x-ray images. In *Computer Vision, Virtual Reality and Robotics in Medicine*, pages 385–391. Springer, 1995.

[19] Charles Soussen and Ali Mohammad-Djafari. Polygonal and polyhedral contour reconstruction in computed tomography. *IEEE Transactions on Image Processing*, 13(11):1507–1523, 2004.

[20] Jovan G Brankov, Yongyi Yang, and Miles N Wernick. Tomographic image reconstruction based on a content-adaptive mesh model. *IEEE Transactions on Medical Imaging*, 23(2):202–212, 2004.

[21] Huaxiang Wang, Lei Tang, and Zhang Cao. An image reconstruction algorithm based on total variation with

adaptive mesh refinement for ECT. *Flow Measurement and Instrumentation*, 18(5):262–267, 2007.

[22] Fanny Buyens, Michele Quinto, and Dominique Houzet. Adaptive mesh reconstruction in x-ray tomography. In *MICCAI Workshop on Mesh Processing in Medical Image Analysis*, 2013.

[23] Daiki Yamanaka, Yutaka Ohtake, and Hiromasa Suzuki. The sinogram polygonizer for reconstructing 3d shapes. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1911–1922, 2013.

[24] Folkert Bleichrodt, Frank Tabak, and Kees Joost Batenburg. Sdart: An algorithm for discrete tomography from noisy projections. *Computer Vision and Image Understanding*, 129:63–74, 2014.

[25] Xiaodong Zhuge, Willem Jan Palenstijn, and Kees Joost Batenburg. Tvr-dart: a more robust algorithm for discrete tomography from limited projection data with automated gray value estimation. *IEEE Transactions on Image Processing*, 25(1):455–468, 2016.

[26] Tim McInerney and Demetri Terzopoulos. T-snakes: Topology adaptive snakes. *Medical image analysis*, 4(2):73–91, 2000.

[27] Marek Krzysztof Misztal and Jakob Andreas Bærentzen. Topology-adaptive interface tracking using the deformable simplicial complex. *ACM Transactions on Graphics (TOG)*, 31(3):24, 2012.

[28] Marek Krzysztof Misztal, Kenny Erleben, Adam Bargteil, Jens Fursund, Brian Bunch Christensen, Jakob Andreas Bærentzen, and Robert Bridson. Multiphase flow of immiscible fluids on unstructured moving meshes. *IEEE transactions on visualization and computer graphics*, 20(1):4–16, 2014.

[29] Vedrana Andersen Dahl, Asger Nyman Christiansen, and Jakob Andreas Bærentzen. Multiphase image segmentation using the deformable simplicial complex method. In *International Conference on Pattern Recognition (ICPR)*, pages 1002–1007. IEEE, 2014.

[30] Andreas Alpers, Richard J Gardner, Stefan König, Robert S Pennington, Chris B Boothroyd, Lothar Houben, Rafal E Dunin-Borkowski, and Kees Joost Batenburg. Geometric reconstruction methods for electron tomography. *Ultramicroscopy*, 128:42–54, 2013.

[31] Chenyang Xu, Dzung L Pham, and Jerry L Prince. Image segmentation using deformable models. *Handbook of medical imaging*, 2:129–174, 2000.

[32] Chenyang Xu, Anthony Yezzi, and Jerry L Prince. On the relationship between parametric and geometric active contours. In *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 483–489. IEEE, 2000.

[33] Anders H Andersen and Avinash C Kak. Simultaneous algebraic reconstruction technique (SART): a superior implementation of the art algorithm. *Ultrasonic Imaging*, 6(1):81–94, 1984.

[34] Wim van Aarle, Willem Jan Palenstijn, Jan De Beenhouwer, Thomas Altantzis, Sara Bals, K. Joost Batenburg, and Jan Sijbers. The ASTRA toolbox: A platform for advanced algorithm development in electron tomography. *Ultramicroscopy*, 157:35 – 47, 2015.

[35] Wim van Aarle, Willem Jan Palenstijn, Jeroen Cant, Eline Janssens, Folkert Bleichrodt, Andrei Dabravolski, Jan De Beenhouwer, K. Joost Batenburg, and Jan Sijbers. Fast and flexible x-ray tomography using the ASTRA toolbox. *Optics Express*, 24(22):25129–25147, Oct 2016.

[36] Per Christian Hansen and Maria Saxild-Hansen. AIR toolsa Matlab package of algebraic iterative reconstruction methods. *Journal of Computational and Applied Mathematics*, 236(8):2167–2178, 2012.